



## Probabilistic Twin Support Vector Machine for Solving Unclassifiable Region Problem

J. A. Nasiri<sup>a</sup>, H. Shakibian<sup>\*b</sup>

<sup>a</sup> Faculty of Mathematical Sciences, Ferdowsi University of Mashhad, Mashhad, Iran

<sup>b</sup> Department of Computer Engineering, Faculty of Engineering, Alzahra University, Tehran, Iran

### PAPER INFO

#### Paper history:

Received 25 August 2021

Received in revised form 07 October 2021

Accepted 11 October 2021

#### Keywords:

Probabilistic Twin Support Vector Machine

Unclassifiable Region

Multi-class Classification

Human Action Recognition

### ABSTRACT

Support Vector Machine classifiers are widely used in many classification tasks. However, they have two considerable weaknesses, Unclassifiable Region (UR) in multiclass classification and outliers. In this research, we address these problems by introducing Probabilistic Least Square Twin Support Vector Machine (PLS-TSVM). The proposed algorithm introduces continuous and probabilistic outputs over the model obtained by Least-Square Twin Support Vector Machine (LS-TSVM) method with both linear and polynomial kernel functions. PLS-TSVM not only solves the unclassifiable region problem by introducing a continuous output value membership function, but it also reduces the adverse effects of noisy data and outliers. For showing the superiority of our proposed method, we have conducted experiments on various UCI datasets. In the most cases, higher or competitive accuracy to other methods have been obtained such that in some UCI datasets, PLS-TSVM could obtain up to 99.90% of classification accuracy. Moreover, PLS-TSVM has been evaluated against "one-against-all" and "one-against-one" approaches on several well-known video datasets such as Weizmann, KTH, and UCF101 for human action recognition task. The results show the higher accuracy of PLS-TSVM compared to its counterparts. Specifically, the proposed algorithm could improve respectively about 12.2%, 2.8%, and 12.1% of classification accuracy in three video datasets compared to the standard SVM and LS-TSVM classifiers. The final results indicate that the proposed algorithm could achieve better overall performances than the literature.

doi: 10.5829/ije.2022.35.01a.01

## 1. INTRODUCTION

Pattern recognition and classification methods are applied to a vast range of real-world applications such as image classification [1-3], disease detection [4], text classification [5], and so on. The significant growth in these applications shows the necessity of fast and classifiers.

For classification tasks, different machine learning classifiers have been applied. The recent research indicates that Support Vector Machines has better performance among other classifiers in most cases [6-11]. However, SVM-based classifiers suffer from several major problems. The first problem is that they are very sensitive to noisy data and outliers. The reason is optimal hyperplane obtained by only a small part of samples (support vectors) [12]. Second, the class-boundary-skew

will be met when SVM is applied to the problem of learning from imbalanced datasets when for instance, one class (suppose negative data) strongly outnumber the other class (positive samples) [13]. In that case, the class boundary can be skewed towards the Negative class. As a result, the false-negative rate can be very high and can make SVM ineffective in identifying the targets that belong to the positive class, which results in the class-boundary-skew problem. Third, the occurrence of unclassifiable regions (URs) when SVM classifiers are applied to multiclass problems.

UR occurs when a multiclass classification application assigns two or more classes to one data sample. In another word, it appears when the classifier can not distinguish between the actual class corresponding to the data-sample and other classes. This may also happen as a result of an imbalance dataset or

\*Corresponding Author Institutional Email: [h.shakibian@alzahra.ac.ir](mailto:h.shakibian@alzahra.ac.ir) (H. Shakibiani)

general noises in image recognition tasks. For instance, suppose an application has to distinguish between daily human activities from a set of short videos. The case study videos may suffer from problems such as viewpoint change, ambient occlusion, illumination change, and intra-class action variations [14] that generally occur in image and video classification tasks. Consequently, the application may face difficulty in discriminating between two similar classes, such as "walking" and "running", and it results in an inaccurate classification.

The problem of unclassifiable regions has been addressed in a number of studies. A weighted voting mechanism formulated by a base classifier has been proposed to eliminate the unclassifiable area [15]. The results based on the logistic regression and support vector machine show higher accuracy and efficiency by modifying the basic classifier. In some of the works, an optimized continuous decision function based on fuzzy support vector machine (FSVM) has been developed to enhance the classification performance [16-18]. A truncated polyhedral pyramidal membership function has been proposed over the decision functions obtained by training the SVM for each class [19]. As the methods obtain the same classification results, for the data points within the classifiable regions, the generalization ability of the FSVM is the same with or better than that of the SVM. A fuzzy classifier based on the support vector machine has been proposed by introducing the concepts of fuzzy linear separability and fuzzy hyperplanes. The proposed fuzzy classifier is robust to the class label perturbation and have been applied in the medical diagnosis. It can obtain a good generalization performance and meanwhile can solve the unclassifiable regions by finding the membership that an example belongs to the positive or negative class [20]. A decision margin based fuzzy output SVM approach has been proposed by Yang [21] to reduce the unclassifiable regions and improve the accuracy of the incident ticket classification. The multiclass support vector machine (SVM) has also been used following decision-directed acyclic graph (DDAG) to address the unclassifiable regions for predicting the unknown fault prediction [22]. Nesting-One-Against-One algorithm [23] and vector projection method [24] are the other methods to handle the unclassifiable region problem.

The proposed algorithms to solve the UR problem are mainly based on the standard SVM algorithm and there are a few corresponding works on the TWSVM. Moreover, since the TWSVM and the SVM have different mechanisms, the approach of modifying the latter cannot be used directly to the former one. Therefore, we aimed to study the UR problem of TWSVM-type algorithms. Accordingly, following the line of research conducted by Khemchandani and Suresh [25], Liu and Yen-Ting [26], a novel Probabilistic Output

Multi-class Least Square Twin Support Vector Machine (PLS-TSVM) algorithm was proposed in which a continuous decision function has been introduced to the outputs of the LS-TSVM hyperplane [27].

To validate our proposed method, we conducted experiments on a set of UCI datasets to compare PLS-TSVM with SVM, TWSVM, and LS-TSVM. Then, PLS-TSVM has been applied on three well-known and widely-used video datasets, namely Weizmann, KTH and UCF101 to compare it with the literature.

## 2. PROBABILISTIC LEAST SQUARE TWIN SUPPORT VECTOR MACHINE

Although much research based on least square twin support vector machine has been presented [28-33] they are incapable of eliminating the consequences of unclassifiable regions. Therefore as it was also mentioned previously, our motivation in this research is to propose a classifier based on LS-TSVM that addresses the unclassifiable regions (URs).

The proposed algorithm consists of two main steps. Firstly, two nonparallel separating hyperplanes are obtained as the solutions of the Quadratic Programming Problems (QPPs) problems in TSVMs. In this step, we introduced both linear and nonlinear models. Secondly, the classification decision rule is defined probabilistically according to the linear and nonlinear models, respectively. In following, we present PLS-TSVM in more details.

### 2.1. Linear PLS-TSVM

Given a dataset  $D$ , we denote the training data of class +1 and class -1 by matrix  $A$  and  $B$ , respectively. If  $d^+$  and  $d^-$  denote the training sets with label +1 and -1, respectively, then  $A$  would be  $\in R^{d^+ \times m}$  and matrix  $B \in R^{d^- \times m}$ , where  $m$  is the problem dimension.

In PLS-TSVM, the primal problem as well as the equality constraints are the same as that of LS-TSVM [27]:

$$\begin{aligned} \text{Min}_{w_{(1)}, b_{(1)}} \quad & \frac{1}{2} \|Aw_{(1)} + eb_{(1)}\|^2 + \frac{c_1}{2} y^T y \\ \text{s.t.} \quad & -(Bw_{(1)} + eb_{(1)}) + y = e \end{aligned} \quad (1)$$

where  $c_1 > 0$  are penalty parameters,  $e$  is the  $\mathbf{1}$ 's vectors, and  $y$  is vectors of slack variables, respectively. Substituting the equality constraints into Equation (1), we have:

$$\text{Min}_{w_{(1)}, b_{(1)}} \quad \frac{1}{2} \|Aw_{(1)} + eb_{(1)}\|^2 + \frac{c_1}{2} \|Bw_{(1)} + eb_{(1)} + e\|^2 \quad (2)$$

Next, the gradient of Equation (2) with respect to  $(w)_{(1)}$  and  $(b)_{(1)}$  is set to zero and we have:

$$A^T (Aw_{(1)} + eb_{(1)}) + c_1 B^T (Bw_{(1)} + eb_{(1)} + e) = 0e, \quad (3)$$

$$e^T (Aw_{(1)} + eb_{(1)}) + c_1 e^T (Bw_{(1)} + eb_{(1)} + e) = 0e, \quad (4)$$

Now, arranging Equations (3) and (4) in matrix form and solving for  $(w)_{(1)}$  and  $(b)_{(1)}$  gives:

$$c_1 \begin{bmatrix} B^T B & B^T e \\ e^T B & e_2 \end{bmatrix} \begin{bmatrix} w_{(1)} \\ b_{(1)} \end{bmatrix} + \begin{bmatrix} A^T A & A^T e \\ e^T A & e_1 \end{bmatrix} \begin{bmatrix} w_{(1)} \\ b_{(1)} \end{bmatrix} + c_1 \begin{bmatrix} B^T \\ e_2 \end{bmatrix} = 0e \quad (5)$$

$$\begin{bmatrix} w_{(1)} \\ b_{(1)} \end{bmatrix} = \begin{bmatrix} c_1 B^T B + A^T A & c_1 B^T e + A^T e \\ c_1 e^T B + e^T A & c_1 m_2 + m_1 \end{bmatrix}^{-1} \times \begin{bmatrix} -c_1 B^T \\ -c_1 e^T \end{bmatrix} \quad (6)$$

$$\begin{bmatrix} w_{(1)} \\ b_{(1)} \end{bmatrix} = - \begin{bmatrix} c_1 \begin{bmatrix} B^T \\ e^T \end{bmatrix} [B e] + \begin{bmatrix} A^T \\ e^T \end{bmatrix} [A e] \end{bmatrix}^{-1} \times \begin{bmatrix} c_1 \begin{bmatrix} B^T \\ e^T \end{bmatrix} \end{bmatrix} \quad (7)$$

Lets  $P = [A e]$  and  $Q = [B e]$ , the solution becomes as:

$$\begin{bmatrix} w_{(1)} \\ b_{(1)} \end{bmatrix} = - [c_1 Q^T Q + P^T P]^{-1} c_1 Q^T e. \quad (8)$$

Similarly, the solution of QPP Equation (9) can be found as shown in Equation (10):

$$\begin{aligned} \text{Min}_{w_{(2)}, b_{(2)}} \quad & \frac{1}{2} \|Bw_{(2)} + eb_{(2)}\|^2 + \frac{c_2}{2} y^T y \\ \text{s.t.} \quad & (Aw_{(2)} + eb_{(2)}) + y = e, \end{aligned} \quad (9)$$

$$\begin{bmatrix} w_{(2)} \\ b_{(2)} \end{bmatrix} = [c_2 P^T P + Q^T Q]^{-1} c_2 P^T e. \quad (10)$$

In this way, two nonparallel separating hyperplanes are obtained. As we know the linear LS-TSVM completely solves the classification problem with just two systems of linear equation as opposed to solving two QPPs in TSVM or one in QPP in SVM which helps the proposed PLS-TSVM to be faster than the other two algorithm in the training phase.

## 2. 2. Nonlinear PLS-TSVM

To obtain the nonlinear model, the following kernel generated surfaces are introduced:

$$K(x^T, C^T) u_{(1)} + \gamma_{(1)} = 0 \quad (11)$$

$$K(x^T, C^T) u_{(2)} + \gamma_{(2)} = 0$$

where  $C = [A; B]$  and  $K$  is an arbitrary kernel. Now, the primal problems of the nonlinear PLS-TSVM is defined with 2-norm of slack variables. The equality constraints corresponding to surfaces Equation (11) are given in Equations (12) and (13), respectively.

$$\begin{aligned} \text{Min}_{u_{(1)}, \gamma_{(1)}} \quad & \frac{1}{2} \|K(A, C^T) u_{(1)} + e\gamma_{(1)}\|^2 + \frac{c_1}{2} y^T y \\ \text{s.t.} \quad & - (K(B, C^T) u_{(1)} + e\gamma_{(1)}) + y = e \end{aligned} \quad (12)$$

and:

$$\begin{aligned} \text{Min}_{u_{(2)}, \gamma_{(2)}} \quad & \frac{1}{2} \|K(B, C^T) u_{(2)} + e\gamma_{(2)}\|^2 + \frac{c_2}{2} y^T y \\ \text{s.t.} \quad & (K(A, C^T) u_{(2)} + e\gamma_{(2)}) + y = e \end{aligned} \quad (13)$$

If the constraints are substituted into the objective function, the QPPs take the following form:

$$\begin{aligned} \text{Min}_{u_{(1)}, \gamma_{(1)}} \quad & \frac{1}{2} \|K(A, C^T) u_{(1)} + e\gamma_{(1)}\|^2 + \\ & \frac{c_1}{2} \|K(B, C^T) u_{(1)} + e\gamma_{(1)} + e\|^2 \end{aligned} \quad (14)$$

$$\begin{aligned} \text{Min}_{u_{(2)}, \gamma_{(2)}} \quad & \frac{1}{2} \|K(B, C^T) u_{(2)} + e\gamma_{(2)}\|^2 + \\ & \frac{c_2}{2} \|-K(A, C^T) u_{(2)} - e\gamma_{(2)} + E_{(2)}\|^2 \end{aligned} \quad (15)$$

Finally, the solutions of Equations (14) and (15) is derived as:

$$\begin{bmatrix} u_{(1)} \\ \gamma_{(1)} \end{bmatrix} = - [c_1 N^T N + M^T M]^{-1} [c_1 N^T e] \quad (16)$$

$$\begin{bmatrix} u_{(2)} \\ \gamma_{(2)} \end{bmatrix} = [c_2 M^T M + N^T N]^{-1} [c_2 M^T e] \quad (17)$$

where  $M = [K(A, C^T)e]$  and  $N = [K(B, C^T); e]$

## 2. 3. Classification Decision Rule

The second step of PLS-TSVM is defining a decision rule for each of the proposed linear and nonlinear model. For a new testing point  $x$ , the corresponding class label is assigned by the following decision function in the linear case where  $i \in \{1, 2, \dots, k\}$ :

$$f_i(x) = \begin{cases} +1 & \text{if } \left| \frac{xw_{(1)} + eb_{(1)}}{xw_{(2)} + eb_{(2)}} \right| \leq 1 \\ 0 & \text{if } \left| \frac{xw_{(1)} + eb_{(1)}}{xw_{(2)} + eb_{(2)}} \right| > 1 \end{cases} \quad (18)$$

In nonlinear PLS-TSVM, the function  $f_i(\cdot)$  is defined as follow:

$$f_i(x) = \begin{cases} +1 & \text{if } \left| \frac{K(x, C^T) u_{(1)} + e\gamma_{(1)}}{K(x, C^T) u_{(2)} + e\gamma_{(2)}} \right| \leq 1 \\ 0 & \text{if } \left| \frac{K(x, C^T) u_{(1)} + e\gamma_{(1)}}{K(x, C^T) u_{(2)} + e\gamma_{(2)}} \right| > 1 \end{cases} \quad (19)$$

The class which earns the highest voting number will be the final output. If more than one class earns the highest voting number, probabilistic output function Equations (20) and (21) will be used to convert the discrete output to the continuous case. Let  $n$  present index of classes earn the highest voting number. In other words, when the decision function outputs the same highest numbers for various classes, it results in unclassifiable regions arisen. Therefore, Equations (20) or (21), will be used for each

individual, and the final decision will be made by comparing the output numbers of the probabilistic decision function Equation (22).

In the case of linear PLS-TSVM, the corresponding decision function is defined as follow  $n \in \{argmax(f_i(x))\}$ :

$$f_n(x) = \begin{cases} M_n(x) & \text{if } \left| \frac{xw_{(1)}+eb_{(1)}}{xw_{(2)}+eb_{(2)}} \right| \leq 1 \\ 0 & \text{if } \left| \frac{xw_{(1)}+eb_{(1)}}{xw_{(2)}+eb_{(2)}} \right| > 1 \end{cases} \quad (20)$$

And in the case of nonlinear PLS-TSVM, the decision function is as follows:

$$f_n(x) = \begin{cases} M_n(x) & \text{if } \left| \frac{K(x,C^T)u_{(1)}+e\gamma_{(1)}}{K(x,C^T)u_{(2)}+e\gamma_{(2)}} \right| \leq 1 \\ 0 & \text{if } \left| \frac{K(x,C^T)u_{(1)}+e\gamma_{(1)}}{K(x,C^T)u_{(2)}+e\gamma_{(2)}} \right| > 1 \end{cases} \quad (21)$$

where the membership function is computed as follow:

$$M_n(x) = \frac{1}{1 + e^{-aD_n(x)+b}} \quad (22)$$

$$D_n(x) = ||xw_{(1)} + eb_{(1)}| - |xw_{(2)} + eb_{(2)}|| \quad (23)$$

Similar, for the nonlinear PLS-TSVM we have:

$$D_n(x) = ||K(x,C^T)u_{(1)} + e\gamma_{(1)}| - |K(x,C^T)u_{(2)} + e\gamma_{(2)}|| \quad (24)$$

The final classification decision of the improved voting strategy is made with Equation (22). Figure 1(a). shows the decision function of LS-TSVM and Figure 1(b). shows  $f(x)$  in PLS-TSVM calculated by Equation (20). for positive and negative samples, the farther from the separating hyperplanes, the greater probabilistic output value  $|f(x)|$  will be obtained. This is to say that  $f(x)$  is a suitable continuous output for LS-TSVM. Similar to the continuous output in SVM and TWSVM, the values range of  $M(x)$  in our model also from negative infinity to positive infinity [34]. In Figure 2, the proposed algorithm has been illustrated within the human action recognition framework.

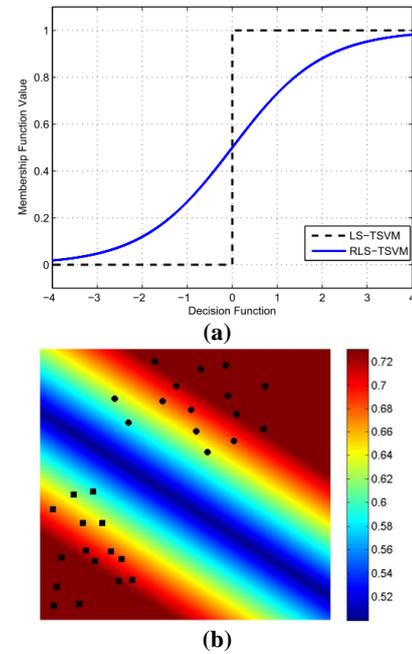
**2. 4. Discussion on PLS-TSVM** PLS-TSVM is an extension of LS-TSVM based on a probabilistic output function to solve unclassifiable regions (URs) problem in multiclass classification.

The constraints of LS-TSVM require the hyperplane to be at a distance of exactly one from the points of the other class. In PLS-SVM, a continuous probabilistic output has been defined. The samples in URs are handled with different degrees of memberships for different classes. We selected the UCI Wine dataset with two of its features to show the effect of PLS-TSVM's decision

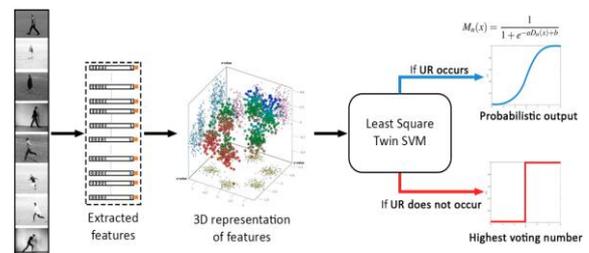
function. Figure 3 shows the proposed PLS-TSVM classifier with its probabilistic decision function within the human action recognition framework based on the one-against-one strategy. Linear kernel and polynomial kernel of degree 4 have been used in this figure. The illustrated black region does not belong to any class (URs). We can observe that the decision function of PLS-TSVM has a high effect on resolving the unclassified region.

**3. EXPERIMENTAL RESULTS**

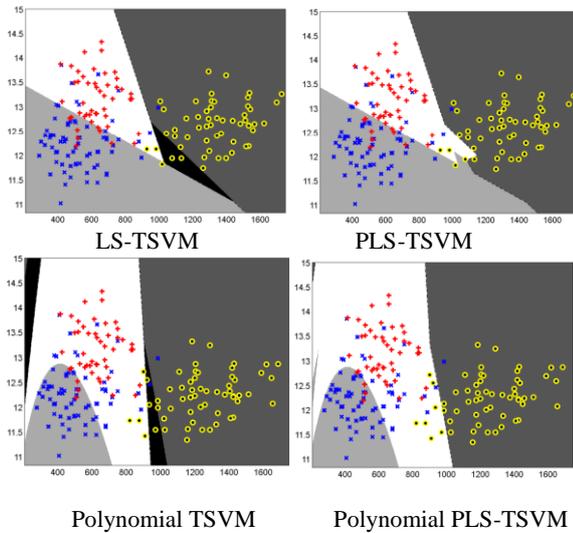
In this section, we evaluated the test results of PLS-TSVM's experiments on UCI datasets. Then we applied PLS-TSVM on human activity recognition application. We first compared the accuracy of PLS-TSVM with



**Figure 1.** (a) indicates decision function of LS-TSVM, (b) illustrates  $f(x)$  calculated by Equation (20) in PLS-TSVM



**Figure 2.** The human action recognition framework based on the proposed algorithm, PLS-TSVM is illustrated in this figure



**Figure 3.** Geometric interpretation of multi-class LS-TSVM and PLS-TSVM

SVM, TWSVM, and LS-TSVM, then compared the obtained results from video datasets, namely, Weizmann, KTH, UCF101 with the literature to show the good performance of PLS-TSVM. The experiments were performed on an Intel Core i7 processor with 32 GB RAM. The optimal values for the parameters were found by the grid search method. In this regard, the optimal values for  $C$  and  $\gamma$  parameters were selected from the range  $\{2^i \mid i = -5, -3, -1, 0, 1, 3, 5\}$ .

### 3. 1. UCI Data Sets

In this subsection, the performance of PLS-TSVM is compared with LS-TSVM, TSVM, and SVM. For these experiments, we utilized 9 UCI data sets, which their 141 characteristics are provided in Table 1. The results of these experiments with Linear and RBF kernels are provided in Tables 2 and 3, respectively, using 5-fold cross-validation method. Optimal  $C$ ,  $\gamma$  parameters were also provided in both tables. In the Tables 2 and 3, the third value in each cell shows the rank of each algorithm based on the corresponding dataset. For example, PLS-TSVM has the rank of 3 in terms of the prediction accuracy among the four SVM-based algorithms using Balance dataset. The average of these ranks have been reported as the overall rank in the last row of the Tables 2 and 3 [30]. As can be seen in both tables, our proposed method obtained the least rank score among the three other algorithms.

As it can be observed in Table 2, PLS-TSVM outperformed other competitors in 6 data sets out of 9. It should also be pointed out that in cases which UR has not occurred, PLS-TSVM has obtained the same accuracy with that of LS-TSVM in both kernels. The results in Table 3 show that our proposed algorithm had better or equal performance in 7 data sets. It can be concluded that PLS-TSVM was successful in the case of improving the

performance of classification in the face of the Unclassifiable region's occurrence.

### 3. 2. Human Action Video Datasets

The reason why we employ the human activity datasets is that in human action recognition (HAR), a strong occurrence of outliers is highly probable due to the errors in key-point detection, noisy data, occlusion, etc. However, there are no capabilities in SVM, TSVM, and LS-TSVM to handle it. The other problem is that these algorithms were originally designed for binary classification, while HAR is practically a problem of multiclass classification. In SVM and TWSVM family framework, "one-against-all" and "one-against-one" approaches are usually solve multiclass classification. They suffer from the unclassifiable region (UR) problem. However, PLS-TSVM resolved unclassifiable region (UR) problem by continuous decision function.

According to the above reasons, PLS-TSVM has been employed to understand human actions. For this purpose, we have compared our PLS-TSVM method with other related methods on the Weizmann, KTH, and UCF101 action datasets.

Figures 4 and 5 provide some sample frames of action datasets. In our experiments, the leave-one-out cross-validation approach was used on Weizmann and KTH to evaluate the performance of the proposed method. In UCF101 experiments, we have used the predefined splits by authors for training-testing and report the average accuracy. The linear kernel has been utilized in all experiments. we set  $a = b = 1$  in membership function to reduce the computational complexity of parameter selection.

#### 3. 2. 1. Weizmann Dataset

In this dataset, there are 90 low-resolution ( $180 \times 144$  pixels) video sequences from 10 natural actions performed by nine persons. The actions (classes) are: walking (walk), running (run), jumping (jump), galloping sideways (side), bending

**TABLE 1.** Characteristics of selected UCI data sets for the experiments

Dataset	# of features	# of features	# of classes
Balance	626	4	3
Dermatology	367	34	6
Glass	215	10	6
Ecoli	328	7	5
Iris	151	4	3
Teaching Evaluation	152	5	3
Wine	179	13	3
Vowels	991	13	10
Vehicle	847	18	4

**TABLE 2.** Experiment results on UCI datasets with Linear Kernel. The optimal values for the parameters  $(C_1, C_2)$  have been found using grid search method in the range of  $\{2^i | i = -5, -3, -1, 0, 1, 3, 5\}$ . The accuracy rank of each algorithm has also been computed and averaged in the last row

Dataset	SVM	TSVM	LS-TSVM	PLS-TSVM
	$(C)$	$(C_1, C_2)$	$(C_1, C_2)$	$(C_1, C_2)$
	Acc $\pm$ std Rank	Acc $\pm$ std Rank	Acc $\pm$ std Rank	Acc $\pm$ std Rank
	$2^{-1}$	$2^{-3}, 2^2$	$2^{-2}, 2^{-5}$	$2^{-2}, 2^{-5}$
<b>Balance</b>	91.70% $\pm$ 0.02% 1	90.72% $\pm$ 1.93% 2	89.68% $\pm$ 3.34% 3	89.68% $\pm$ 3.34% 3
	$2^{-3}$	$2^{-4}, 2^{-2}$	$2^3, 2^3$	$2^5, 2^3$
<b>Dermatology</b>	97.80% $\pm$ 0.01% 2	97.56% $\pm$ 3.66% 3	97.60% $\pm$ 3.25% 3	98.44% $\pm$ 2.98% 1
	$2^4$	$2^{-2}, 2^{-1}$	$2^{-3}, 2^{-3}$	$2^{-3}, 2^{-3}$
<b>Ecoli</b>	89.60% $\pm$ 0.02% 3	89.58% $\pm$ 3.80% 4	89.67% $\pm$ 3.69% 2	89.71% $\pm$ 3.40% 1
	$2^2$	$2^2, 2^{-5}$	$2^{-3}, 2^{-4}$	$2^{-3}, 2^{-4}$
<b>Glass</b>	95.30% $\pm$ 0.03% 1	91.12% $\pm$ 3.98% 4	93.94% $\pm$ 5.00% 3	94.15% $\pm$ 3.81% 2
	$2^{-3}$	$2^{-2}, 2^{-3}$	$2^{-2}, 2^{-3}$	$2^{-2}, 2^{-3}$
<b>Iris</b>	98.04% $\pm$ 0.04% 1	98.00% $\pm$ 1.63% 2	98.00% $\pm$ 1.63% 2	98.00% $\pm$ 1.63% 2
	$2^4$	$2^{-2}, 2^{-1}$	$2^{-3}, 2^{-3}$	$2^{-3}, 2^{-3}$
<b>Optdigits</b>	89.64% $\pm$ 0.02% 3	89.36% $\pm$ 3.77% 4	89.78% $\pm$ 3.15% 2	90.18% $\pm$ 3.40% 1
	$2^1$	$2^1, 2^0$	$2^{-2}, 2^{-2}$	$2^{-2}, 2^{-2}$
<b>Teaching Evaluation</b>	56.30% $\pm$ 0.09% 3	55.65% $\pm$ 9.40% 1	53.05% $\pm$ 11.61% 4	56.35% $\pm$ 11.54% 2
	$2^{-5}$	$2^{-1}, 2^{-4}$	$2^{-4}, 2^{-3}$	$2^{-4}, 2^{-3}$
<b>Wine</b>	98.90% $\pm$ 0.02% 1	97.76% $\pm$ 2.08% 2	98.73% $\pm$ 2.13% 3	98.73% $\pm$ 2.13% 3
	$2^{-4}$	$2^{-1}, 2^{-1}$	$2^{-2}, 2^{-2}$	$2^{-2}, 2^{-2}$
<b>Vehicle</b>	81.10% $\pm$ 0.01% 4	81.20% $\pm$ 1.11% 3	81.56% $\pm$ 1.99% 2	82.74% $\pm$ 1.80% 1
	$2^4$	$2^{-2}, 2^{-1}$	$2^0, 2^{-1}$	$2^0, 2^{-1}$
<b>Vowel</b>	82.60% $\pm$ 0.03% 2	75.45% $\pm$ 2.42% 4	76.96% $\pm$ 2.84% 3	83.02% $\pm$ 2.48% 1
<b>Overall Rank</b>	2.33	3.33	3.00	1.88

(bend), one-hand-waving (wave1), two-hands-waving (wave2), jumping in place (pjump), jumping jack (jack), and skipping (skip). To compute the recognition rates,



(a)



(b)

**Figure 4.** Some examples of video sequences in (a) Weizmann and (b) KTH datasets

**TABLE 3.** Experiment results on UCI datasets with RBF Kernel. The optimal values for the parameters  $(C_1, C_2, \gamma)$  have been found using grid search method in the range of  $\{2^i | i = -5, -3, -1, 0, 1, 3, 5\}$ . The accuracy rank of each algorithm has also been computed and averaged in the last row.

Dataset	SVM ( $C, \gamma$ ) Acc $\pm$ std Rank	TSVM ( $C_1, C_2, \gamma$ ) Acc $\pm$ std Rank	LS-TSVM ( $C_1, C_2, \gamma$ ) Acc $\pm$ std Rank	PLS-TSVM ( $C_1, C_2, \gamma$ ) Acc $\pm$ std Rank
<b>Balance</b>	$2^3, 2^{-3}$ 96.18 % $\pm$ 0.02 % 4	$2^{-3}, 2^{-1}, 2^{-5}$ 95.36 % $\pm$ 0.78 % 3	$2^{-3}, 2^{-3}, 2^{-5}$ 99.52 % $\pm$ 0.39 % 2	$2^{-3}, 2^{-3}, 2^{-5}$ 99.84 % $\pm$ 0.41 % 1
<b>Dermatology</b>	$2^{-1}, 2^{-1}$ 91.70 % $\pm$ 0.02 % 1	$2^{-3}, 2^2, 2^{-1}$ 90.72 % $\pm$ 1.93 % 2	$2^{-2}, 2^{-5}, 2^{-1}$ 89.68 % $\pm$ 3.34 % 3	$2^{-2}, 2^{-5}, 2^{-1}$ 89.68 % $\pm$ 2.74 % 3
<b>Ecoli</b>	$2^{-1}, 2^2$ 88.00 % $\pm$ 0.00 % 3	$2^{-3}, 2^{-1}, 2^{-4}$ 89.89 % $\pm$ 1.93 % 1	$2^{-3}, 2^{-1}, 2^{-4}$ 89.29 % $\pm$ 2.55 % 1	$2^{-3}, 2^{-1}, 2^{-4}$ 89.29 % $\pm$ 2.55 % 1
<b>Glass</b>	$2^4, 2^{-4}$ 93.00 % $\pm$ 0.00 % 4	$2^{-5}, 2^4, 2^{-5}$ 97.19 % $\pm$ 1.75 % 3	$2^{-2}, 2^{-5}, 2^{-1}$ 99.44 % $\pm$ 1.11 % 2	$2^{-2}, 2^{-5}, 2^{-1}$ 99.90 % $\pm$ 1.11 % 1
<b>Iris</b>	$2^2, 2^{-4}$ 97.00 % $\pm$ 0.00 % 3	$2^{-4}, 2^{-5}, 2^{-5}$ 98.00 % $\pm$ 1.63 % 2	$2^{-1}, 2^{-1}, 2^4$ 98.66 % $\pm$ 1.63 % 1	$2^{-1}, 2^{-1}, 2^4$ 98.66 % $\pm$ 1.63 % 1
<b>Optdigits</b>	$2^1, 2^{-5}$ 99.28 % $\pm$ 0.00 % 2	$2^{-4}, 2^{-4}, 2^{-5}$ 98.89 % $\pm$ 1.00 % 3	$2^0, 2^{-2}, 2^{-5}$ 99.50 % $\pm$ 0.59 % 1	$2^0, 2^{-2}, 2^{-5}$ 99.50 % $\pm$ 0.59 % 1
<b>Teaching Evaluation</b>	$2^5, 2^2$ 60.30 % $\pm$ 0.02 % 4	$2^{-2}, 2^{-1}, 2^{-1}$ 66.79 % $\pm$ 8.37 % 1	$2^0, 2^0, 2^{-4}$ 62.50 % $\pm$ 10.85 % 3	$2^0, 2^0, 2^{-4}$ 64.50 % $\pm$ 9.99 % 1
<b>Wine</b>	$2^{-1}, 2^{-5}$ 98.90 % $\pm$ 0.02 % 2	$2^{-5}, 2^{-4}, 2^{-5}$ 97.19 % $\pm$ 1.75 % 3	$2^0, 2^{-1}, 2^{-4}$ 99.44 % $\pm$ 1.12 % 1	$2^0, 2^{-1}, 2^{-4}$ 99.44 % $\pm$ 1.12 % 1
<b>Vehicle</b>	$2^5, 2^{-5}$ 82.51 % $\pm$ 0.04 % 4	$2^{-3}, 2^{-3}, 2^{-5}$ 83.46 % $\pm$ 2.86 % 3	$2^{-5}, 2^{-4}, 2^{-5}$ 84.99 % $\pm$ 2.29 % 2	$2^{-5}, 2^{-4}, 2^{-5}$ 87.69 % $\pm$ 2.46 % 1
<b>Vowel</b>	$2^{-5}, 2^{-3}$ 99.6 % $\pm$ 0.0 % 1	$2^{-5}, 2^{-3}, 2^{-5}$ 86.96 % $\pm$ 2.12 % 4	$2^4, 2^4, 2^{-5}$ 95.45 % $\pm$ 2.30 % 3	$2^4, 2^4, 2^{-5}$ 95.55 % $\pm$ 2.30 % 1
<b>Overall Rank</b>	3.11	2.77	2.11	1.55

leave-one-out cross-validation has been employed in which the videos of 8 actors are used as the training dataset and one person for the test.

The results of PLS-TSVM has been reported in Table 4 which are the average accuracy rate of 9 independent runs. As it has been shown, the accuracy rate of PLS-TSVM is higher than most of the state-of-the-art methods. However, some approaches, such as [35-41] reported more accurate predictions. In this regard, it should be noted that these later approaches used additional data founded by tracking or background subtraction. Furthermore, the training time of some methods such as Ada-boost based classifiers is considerably higher than PLS-TSVM.



**Figure 5.** Example frames from video sequences of UCF101 datasets

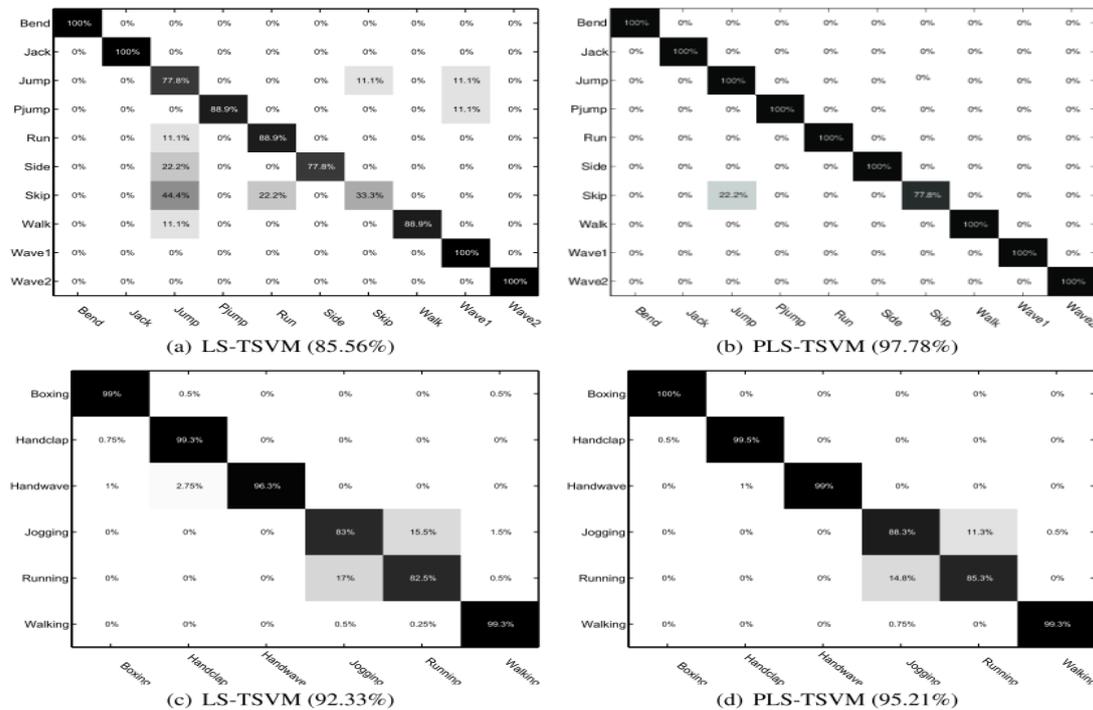
**TABLE 4.** Accuracy rates of different methods on the Weizmann dataset (\*Background subtraction has been used to localize actors)

Method	Classifier	Average Accuracy
Niebles [42]	SVM	72.80%
Liu [43]	KNN	71.69%
Fathi [40]	Adaboost	100%*
Bregonzio [44]	SVM-NN	96.66%
Wang [45]	SVM	92.1 %
Jiang (motion) [41]	Tree Learning+KNN	88.89%*
Jiang (shape) [41]	Tree Learning+KNN	81.11%*
Jiang (shape+motion) [41]	Tree Learning+KNN	100%*
Chou [46]	NNC-GMMC	95.56%
Goudelis [47]	SVM	95.42%
Arunnehru [48]	3D-CNN	96.37%
Nasiri [27]	LS-TSVM	85.56%
Singh [49]	SVM	97.66%
Aslan [50]	KNN	91.11%
Vishwakarma [51]	SVM	97.50 %
Vishwakarma [52]	SVM-HMM	96.00%
Ramya [53]	NN	92.50%
<b>Our method</b>	<b>PLS-TSVM</b>	<b>97.78%</b>

In order to demonstrate the capabilities of PLS-TSVM, the accuracy rate and training time between SVM, LS-TSVM, and PLS-TSVM has also been compared in Table 5. As it is shown, the Haris3D detector and HOG/HOF descriptor are similarly conducted in the experiment. We observe that PLS-TSVM outperformed the LS-TSVM result by 12% on Weizmann. Also, PLS-TSVM performed several orders of magnitude faster than SVM. Figure 6 (a,b) shows the confusion matrices for the Weizmann dataset with the LS-TSVM and PLS-TSVM classifiers, respectively.

**3. 2. 2. KTH Dataset** KTH dataset was introduced in literature [54] which has six types of human actions namely: walking, jogging, running, boxing, hand waving, and hand clapping, performed several times by 25 subjects. Similar to the Weizmann dataset, leave-one-out cross-validation has been employed. The confusion matrix of LS-TSVM and PLS-TSVM has been shown in Figure 6 (c,d). It shows more accurate predictions found by PLS-TSVM in comparison with the LS-TSVM. Overall accuracy is 95.21% on average. The accuracy rates of different methods on the KTH dataset are shown in Table 6.

The computational time of PLS-TSVM has been shown in Table 7 according to one-against-one protocol for multiclass classification. From the table, it is observed that the training time of all leave-one-out cross-validation



**Figure 6.** Confusion matrix of PLS-TSVM compared to LS-TSVM: (a, b) the Weizmann dataset, (c,d) the KTH dataset. me examples of video sequences in (a) Weizmann and (b) KTH datasets

**TABLE 5.** Performance on the Weizmann with Harris3D detector and HOG/HOF descriptor

Classifier	SVM	LS-TSVM	PLS-TSVM
<b>Computational Time</b>	Time(s)	Time(s)	Time(s)
<b>Harris3D+HOG/HOF</b>	84.4%	85.56%	97.78%
	2.9497	0.1082	0.1259

is about 8 minutes and 5 hours for PLS-TSVM and SVM, respectively. This is because PLS-TSVM does not require any special optimizers, whereas SVM has been implemented with fast interior-point solvers of the Mosek optimization toolbox for MATLAB. It is also interesting to mention that using two nonparallel hyperplanes in PLS-TSVM gives an accurate model for human activity.

**TABLE 6.** Accuracy rates of different methods on KTH dataset

Method	Classifier	Average Accuracy
Schuldt [54]	SVM	71%
Dollar [55]	KNN	81%
Wong [56]	WX-SVM	91.6%
Jhuang [57]	Bio-Inspired	91.7%
Niebles [42]	pLSA	83%
Fathi [40]	AdaBoost	90.5%
Klaser [58]	SVM	91.4%
Liu [59]	VWC-Correlation	94.16%
Wang [45]	SVM	92.1 %
Kovashka [60]	SVM	94.53%
Shao [61]	SVM	93.89%
ghodrati [62]	clustering+KNN	93%
Jiang [41]	Tree Learning+KNN	93.4%
Liu [37]	Boosted NBNN	92.7%
Goudelis [47]	SVM	93.14%
Chou [46]	NNC-GMMC	90.58%
An [63]	Deep Model	91.2%
Arunnehru [48]	3D-CNN	93.43%
Nasiri [27]	LS-TSVM	92.33%
Singh [49]	SVM	94.50%
Aslan [50]	KNN	96.14%
Vishwakarma [51]	SVM	96.60 %
Vishwakarma [52]	SVM-HMM	96.66%
Ramya [53]	NN	91.40%
<b>Our method</b>	PLS-TSVM	95.21%

**TABLE 7.** Performance on the KTH with Harris3D detector and HOG/HOF descriptor

Classifier	SVM	LS-TSVM	PLS-TSVM
<b>Computational Time</b>	Time (s)	Time (s)	Time (s)
<b>Harris3D+HOG/</b>	91.8 %	92.33%	95.21%
<b>G/</b>			
<b>HOF</b>	≈18340 (5h)	448.1872 (8min)	443.1907 (8min)

### 3. 2. 3. UCF101

UCF101 is one of the largest realistic datasets for human activity recognition, collected from YouTube [64]. The dataset is composed of 13,320 videos from 101 action categories. It gives the largest diversity in terms of actions in the presence of large variations in camera motion, object appearance and pose, object scale, viewpoint, cluttered background, illumination conditions, etc. Each of the 101 action classes belongs to one of five class types: Human-Object Interaction, Body-Motion Only, Human-Human Interaction, Playing Music Instruments, and Sports (see Figure 5). In these experiments, we have used the predefined splits by authors for training-testing and report the average accuracy.

We measure the overall performance of PLS-TSVM using three standard partitions. Also, the PLS-TSVM classifier is compared to an LS-TSVM classifier in Table 8. We observe that PLS-TSVM outperformed the LS-TSVM results by 12% on UCF101. It seems, PLS-TSVM is a robust classifier that could eliminate the unclassifiable regions (URs) and be more robust in the face of outliers of each class. To further show the advantage of PLS-TSVM with researches that have been published using the same features, we have compared accuracy rates in Table 9. It could be found that the accuracy rate of PLS-TSVM is higher than several state-of-the-art methods. It is worth to mention that we have also provided a variety of deep models in the benchmark tables for all three datasets. However, the comparison of these models with the experimental method used in this paper is unfair since most of the human action recognition methods extract high-level features simultaneously with classification, while in this research, we focused only on the classification performance rather than feature extraction.

**TABLE 8.** Details performance on the UCF101 with Harris3D detector and HOG/HOF descriptor

splits	LS-TSVM	PLS-TSVM
<b>split 1</b>	59.73%	71.75%
<b>split 2</b>	63.50 %	71.21 %
<b>split 3</b>	55.40 %	72.00 %
<b>overall</b>	59.55%	71.66%

**TABLE 9.** Accuracy rates of different methods on UCF101 dataset with HOG/HOF descriptor

Method	Classifier	Average Accuracy
Schuldt [64]	SVM	43.90%
Karpathy [65]	Neural Net	65.40%
Hou [66]	DaMN	57.60%
Boyraz [67]	Neural Net	53.35%
Kihl (baseline HOG) [68]	SVM	65.30%
Kihl (baseline HOF) [68]	SVM	68.60%
Peng [69]	Deep Model	39.94%
Chang [70]	Deep Model	70.94%
Nasiri [27]	LS-TSVM	59.55%
Hua (2D geometry-based) [71]	Deep	62.03%
Francisco [72]	Naive-Bayes	62.03%
Leyva [73]	Fisher Vector	71.60 %
Prakash [74]	Rand. Tree	65.11%
<b>Our method</b>	<b>PLS-TSVM</b>	<b>71.66%</b>

#### 4. CONCLUSION

In this paper, Probabilistic Least Square Twin Support Vector Machine (PLS-TSVM) has been introduced. PLS-TSVM addressed several problems that may occur in TSVM-based algorithms such as unclassifiable regions (URs) and their sensitivity to outliers when they are applied to multiclass classification tasks such as human activity recognition. PLS-TSVM classifier performs classification by the use of two nonparallel hyperplanes similar to TSVM, unlike SVM, which uses a single hyperplane. Finally, a continuous output value is defined by comparing the distances between the samples and two separating hyperplanes to handle URs. In this research, we had two approaches to evaluate our proposed method. We first conducted experiments with PLS-TSVM on a set of UCI data sets and compared the results with SVM, TSVM, and LS-TSVM. Then we applied PLS-TSVM to 3 well-known human action video data sets and provided the results to be able to compare with the literature. For these experiments, we have used the HOG/HOF descriptor to present each video sequence in the bag of words (BoW) model. The results indicate that our proposed PLS-TSVM reaches a better performance on UCI data sets compared to the other three algorithms and also produces a significant improvement in action recognition while the computational time of the method is several orders of magnitude faster than SVM and AdaBoost classification based methods.

#### 5. REFERENCES

- Cen, Feng, Xiaoyu Zhao, Wuzhuang Li, and Guanghui Wang. "Deep feature augmentation for occluded image classification." *Pattern Recognition* 111, (2021), 107737, doi: 10.1016/j.patcog.2020.107737.
- AlyanNezhadi, M. M., H. Qazanfari, A. Ajam, and Z. Amiri. "Content-based Image Retrieval Considering Colour Difference Histogram of Image Texture and Edge Orientation." *International Journal of Engineering, Transactions B: Applications*, Vol. 33, No. 5, (2020), 949-958, doi: 10.5829/ije.2020.33.05b.28.
- Sezavar, A., H. Farsi, and Sajad Mohamadzadeh. "A modified grasshopper optimization algorithm combined with cnn for content-based image retrieval." *International Journal of Engineering, Transactions A: Basics*, Vol. 32, No. 7, (2019), 924-930, doi: 10.5829/ije.2019.32.07a.04.
- Sharma, Parul, Yash Paul Singh Berwal, and Wiqas Ghai. "Performance analysis of deep learning CNN models for disease detection in plants using image segmentation." *Information Processing in Agriculture* 7, No. 4 (2020), 566-574, doi: 10.1016/j.inpa.2019.11.001
- Chen, Long, Liangxiao Jiang, and Chaoqun Li. "Modified DFS-based term weighting scheme for text classification." *Expert Systems with Applications* 168, (2021), 114438, doi: 10.1016/j.eswa.2020.114438
- Rahmanimanesh, Mohammad, Jalal A. Nasiri, Saeed Jalili, and N. Moghaddam Charkari. "Adaptive three-phase support vector data description." *Pattern Analysis and Applications* 22, No. 2, (2019), 491-504, doi: 10.1007/s10044-017-0646-3
- Refahi, Mohammad S., Jalal A. Nasiri, and S. M. Ahadi. "Ecg arrhythmia classification using least squares twin support vector machines." In *Electrical Engineering (ICEE), Iranian Conference on*, 1619-1623. IEEE, 2018.
- Okwuashi, Onuwa, and Christopher E. Ndehedehe. "Deep support vector machine for hyperspectral image classification." *Pattern Recognition* 103, (2020), 107298, doi: 10.1016/j.patcog.2020.107298
- Sivaram, M., E. Laxmi Lydia, Irina V. Pustokhina, Denis Alexandrovich Pustokhin, Mohamed Elhoseny, Gyanendra Prasad Joshi, and K. Shankar. "An optimal least square support vector machine-based earnings prediction of blockchain financial products." *IEEE Access* 8, (2020), 120321-120330, doi: 10.1109/ACCESS.2020.3005808
- Gao, Zheming, Shu-Cherng Fang, Jian Luo, and Negash Medhin. "A kernel-free double well potential support vector machine with applications." *European Journal of Operational Research* 290, No. 1 (2021), 248-262, doi: 10.1016/j.ejor.2020.10.040
- Badaghei, R., H. Hassanpour, and T. Askari. "Detection of Bikers without Helmet Using Image Texture and Shape Analysis." *International Journal of Engineering, Transactions C: Aspects*, Vol. 34, No. 3 (2021): 650-655, doi: 10.5829/ije.2021.34.03c.09
- Wang, Kuaini, Wenxin Zhu, and Ping Zhong. "Robust support vector regression with generalized loss function and applications." *Neural Processing Letters* 41, No. 1 (2015), 89-106, doi: 10.1007/s11063-013-9336-3
- Qu, Hai-Ni, Guo-Zheng Li, and Wei-Sheng Xu. "An asymmetric classifier based on partial least squares." *Pattern Recognition* 43, No. 10 (2010), 3448-3457, doi: 10.1016/j.patcog.2010.05.002
- Guo, Guodong, and Alice Lai. "A survey on still image based human action recognition." *Pattern Recognition* 47, No. 10 (2014), 3343-3361, doi: 10.1016/j.patcog.2014.04.018
- Xiao, Yanghao, Yucheng Liu, Yuanyuan Deng, and Haoxuan Li. "Enhancing Multi-Class Classification in One-Versus-One Strategy: A Type of Base Classifier Modification and Weighted

- Voting Mechanism." In 2021 International Conference on Communications, Information System and Computer Engineering (CISCE), 303-307. IEEE, 2021.
16. Wu, Yuanyuan, Liyong Shen, and Sanguo Zhang. "Fuzzy multiclass support vector machines for unbalanced data." In 2017 29th Chinese Control and Decision Conference (CCDC), 2227-2231. IEEE, 2017.
  17. Pruengkarn, Ratchakoon, Kok Wai Wong, and Chun Che Fung. "Imbalanced data classification using complementary fuzzy support vector machine techniques and SMOTE." In 2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC), 978-983. IEEE, 2017.
  18. Liu, Jie, and Enrico Zio. "A scalable fuzzy support vector machine for fault detection in transportation systems." *Expert Systems with Applications* 102 (2018), 36-43, doi: 10.1016/j.eswa.2018.02.017
  19. Inoue, Takuya, and Shigeo Abe. "Fuzzy support vector machines for pattern classification." In IJCNN'01. International Joint Conference on Neural Networks. Proceedings (Cat. No. 01CH37222), Vol. 2, 1449-1454. IEEE, 2001.
  20. Ji, Ai-bing, Songcan Chen, and Qiang Hua. "Fuzzy classifier based on fuzzy support vector machine." *Journal of Intelligent & Fuzzy Systems* 26, No. 1 (2014), 421-430, doi: 10.3233/IFS-130819.
  21. Yang, Libo. "Fuzzy Output Support Vector Machine Based Incident Ticket Classification." *IEICE Transactions on Information and Systems* 104, No. 1 (2021), 146-151, doi: 10.1587/transinf.2020EDP7044
  22. Thakur, Arunava Kabiraj, Palash Kumar Kundu, and Arabinda Das. "Prediction of Unknown Fault of Induction Motor using SVM following Decision-Directed Acyclic Graph." *Journal of The Institution of Engineers (India): Series B*, 102, No. 3, (2021), 573-583, doi: 10.1007/s40031-021-00536-2
  23. Liu, Bo, Zhifeng Hao, and Eric CC Tsang. "Nesting one-against-one algorithm based on SVMs for pattern classification." *IEEE Transactions on Neural Networks* 19, No. 12, (2008), 2044-2052, doi: 10.1109/TNN.2008.2003298.
  24. Li, Renbing, Aihua Li, Tao Wang, and Liang Li. "Vector projection method for unclassifiable region of support vector machine." *Expert Systems with Applications* 38, No. 1, (2011), 856-861, doi: 10.1016/j.eswa.2010.07.046.
  25. Khemchandani, Reshma, and Suresh Chandra. "Twin support vector machines for pattern classification." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29, No. 5, (2007), 905-910, doi: 10.1109/TPAMI.2007.1068
  26. Liu, Yi-Hung, and Yen-Ting Chen. "Face recognition using total margin-based adaptive fuzzy support vector machines." *IEEE Transactions on Neural Networks* 18, No. 1, (2007), 178-192, doi: 10.1109/TNN.2006.883013.
  27. Nasiri, Jalal A., Nasrollah Moghadam Charkari, and Saeed Jalili. "Least squares twin multi-class classification support vector machine." *Pattern Recognition* 48, No. 3, (2015), 984-992, doi: 10.1016/j.patcog.2014.09.020.
  28. Gao, Zheming, Shu-Cherng Fang, Xuerui Gao, Jian Luo, and Negash Medhin. "A novel kernel-free least squares twin support vector machine for fast and accurate multi-class classification." *Knowledge-Based Systems* 226 (2021), 107123, doi: 10.1016/j.knosys.2021.107123.
  29. Xu, Yitian, Xianli Pan, Zhijian Zhou, Zhiji Yang, and Yuqun Zhang. "Structural least square twin support vector machine for classification." *Applied Intelligence* 42, No. 3, (2015), 527-536, doi: 10.1007/s10489-014-0611-4.
  30. Mir, A., and Jalal A. Nasiri. "KNN-based least squares twin support vector machine for pattern classification." *Applied Intelligence* 48, No. 12, (2018), 4551-4564, doi: 10.1007/s10489-018-1225-z.
  31. Nasiri, Jalal A., Nasrollah Moghadam Charkari, and Kourosh Mozafari. "Energy-based model of least squares twin support vector machines for human action recognition." *Signal Processing* 104 (2014), 248-257, doi: 10.1016/j.sigpro.2014.04.010.
  32. Chen, Xiaobo, Jian Yang, Qiaolin Ye, and Jun Liang. "Recursive projection twin support vector machine via within-class variance minimization." *Pattern Recognition* 44, No. 10-11, (2011), 2643-2655, doi: 10.1016/j.patcog.2011.03.001.
  33. Chen, Su-Gen, and Xiao-Jun Wu. "Multiple birth least squares support vector machine for multi-class classification." *International Journal of Machine Learning and Cybernetics* 8, No. 6, (2017), 1731-1742, doi: 10.1007/s13042-016-0554-7.
  34. Shao, Yuan-Hai, Nai-Yang Deng, Zhi-Min Yang, Wei-Jie Chen, and Zhen Wang. "Probabilistic outputs for twin support vector machines." *Knowledge-Based Systems* 33, (2012), 145-151, doi: 10.1016/j.knosys.2012.04.006.
  35. Bottou, Léon, Corinna Cortes, John S. Denker, Harris Drucker, Isabelle Guyon, Larry D. Jackel, Yann LeCun et al. "Comparison of classifier methods: a case study in handwritten digit recognition." In Proceedings of the 12th IAPR International Conference on Pattern Recognition, Vol. 3-Conference C: Signal Processing (Cat. No. 94CH3440-5), Vol. 2, 77-82. IEEE, 1994.
  36. KRESSEL, Ulrich HG. "Pairwise classification and support vector machines." *Advances in Kernel Methods: Support Vector Learning* (2002).
  37. Liu, Li, Ling Shao, and Peter Rockett. "Human action recognition based on boosted feature selection and naive Bayes nearest-neighbor classification." *Signal Processing* 93, No. 6, (2013), 1521-1530, doi: 10.1016/j.sigpro.2012.07.017.
  38. Lu, Zhiwu, and Yuxin Peng. "Latent semantic learning with structured sparse representation for human action recognition." *Pattern Recognition* 46, No. 7, (2013), 1799-1809, doi: 10.1016/j.patcog.2012.09.027.
  39. Laptev, Ivan, Marcin Marszałek, Cordelia Schmid, and Benjamin Rozenfeld. "Learning realistic human actions from movies." In 2008 IEEE Conference on Computer Vision and Pattern Recognition, 1-8. IEEE, 2008.
  40. Fathi, Alireza, and Greg Mori. "Action recognition by learning mid-level motion features." In 2008 IEEE Conference on Computer Vision and Pattern Recognition, 1-8. IEEE, 2008.
  41. Jiang, Zhuolin, Zhe Lin, and Larry Davis. "Recognizing human actions by learning and matching shape-motion prototype trees." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34, No. 3, (2012), 533-547, doi: 10.1109/TPAMI.2011.147.
  42. Niebles, Juan Carlos, Hongcheng Wang, and Li Fei-Fei. "Unsupervised learning of human action categories using spatial-temporal words." *International Journal of Computer Vision* 79, No. 3, (2008), 299-318, doi: 10.1007/s11263-007-0122-4.
  43. Liu, Jingen, Saad Ali, and Mubarak Shah. "Recognizing human actions using multiple features." In 2008 IEEE Conference on Computer Vision and Pattern Recognition, 1-8. IEEE, 2008.
  44. Bregonzio, Matteo, Shaogang Gong, and Tao Xiang. "Recognising action as clouds of space-time interest points." In 2009 IEEE conference on computer vision and pattern recognition, 1948-1955. IEEE, 2009.
  45. Wang, Heng, Muhammad Muneeb Ullah, Alexander Klaser, Ivan Laptev, and Cordelia Schmid. "Evaluation of local spatio-temporal features for action recognition." In Bmvc 2009-british machine vision conference, pp. 124-1. BMVA Press, 2009.
  46. Chou, Kuang-Pen, Mukesh Prasad, Di Wu, Nabin Sharma, Dong-Lin Li, Yu-Feng Lin, Michael Blumenstein, Wen-Chieh Lin, and

- Chin-Teng Lin. "Robust feature-based automated multi-view human action recognition system." *IEEE Access* 6, (2018), 15283-15296, doi: 10.1109/ACCESS.2018.2809552.
47. Goudelis, Georgios, Konstantinos Karpouzis, and Stefanos Kollias. "Exploring trace transform for robust human action recognition." *Pattern Recognition* 46, No. 12, (2013), 3238-3248, doi: 10.1016/j.patcog.2013.06.006.
  48. Arunnehru, J., G. Chamundeswari, and S. Prasanna Bharathi. "Human action recognition using 3D convolutional neural networks with 3D motion cuboids in surveillance videos." *Procedia Computer Science* 133 (2018), 471-477, doi: 10.1016/j.procs.2018.07.059.
  49. Singh, Tej, and Dinesh Kumar Vishwakarma. "A hybrid framework for action recognition in low-quality video sequences." arXiv preprint arXiv:1903.04090 (2019),
  50. Aslan, Muhammet Fatih, Akif Durdu, and Kadir Sabanci. "Human action recognition with bag of visual words using different machine learning methods and hyperparameter optimization." *Neural Computing and Applications* 32, No. 12, (2020), 8585-8597, doi: 10.1007/s00521-019-04365-9.
  51. Vishwakarma, Dinesh Kumar, and Chhavi Dhiman. "A unified model for human activity recognition using spatial distribution of gradients and difference of Gaussian kernel." *The Visual Computer* 35, No. 11, (2019), 1595-1613, doi: 10.1007/s00371-018-1560-4.
  52. Vishwakarma, Dinesh Kumar. "A two-fold transformation model for human action recognition using decisive pose." *Cognitive Systems Research* 61, (2020), 1-13, doi: 10.1016/j.cogsys.2019.12.004.
  53. Ramya, P., and Rajendran Rajeswari. "Human action recognition using distance transform and entropy-based features." *Multimedia Tools and Applications* 80, No. 6, (2021), 8147-8173, doi: 10.1007/s11042-020-10140-z.
  54. Schuldt, Christian, Ivan Laptev, and Barbara Caputo. "Recognizing human actions: a local SVM approach." In Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004., Vol. 3, 32-36. IEEE, 2004.
  55. Dollár, Piotr, Vincent Rabaud, Garrison Cottrell, and Serge Belongie. "Behavior recognition via sparse spatio-temporal features." In 2005 IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance, 65-72. IEEE, 2005.
  56. Wong, Shu-Fai, Tae-Kyun Kim, and Roberto Cipolla. "Learning motion categories using both semantic and structural information." In 2007 IEEE Conference on Computer Vision and Pattern Recognition, 1-6. IEEE, 2007.
  57. Jhuang, Hueihan, Thomas Serre, Lior Wolf, and Tomaso Poggio. "A biologically inspired system for action recognition." In 2007 IEEE 11th international conference on computer vision, 1-8. Ieee, 2007.
  58. Klaser, Alexander, Marcin Marszałek, and Cordelia Schmid. "A spatio-temporal descriptor based on 3d-gradients." In BMVC 2008-19th British Machine Vision Conference, 275-301. British Machine Vision Association, 2008.
  59. Liu, Jingen, and Mubarak Shah. "Learning human actions via information maximization." In 2008 IEEE Conference on Computer Vision and Pattern Recognition, 1-8. IEEE, 2008.
  60. Kovashka, Adriana, and Kristen Grauman. "Learning a hierarchy of discriminative space-time neighborhood features for human action recognition." In 2010 IEEE computer society conference on computer vision and pattern recognition, 2046-2053. IEEE, 2010.
  61. Shao, Ling, Ruoyun Gao, Yan Liu, and Hui Zhang. "Transform based spatio-temporal descriptors for human action recognition." *Neurocomputing* 74, No. 6, (2011), 962-973, doi: 10.1016/j.neucom.2010.11.013.
  62. Ghodrati, Amir, and Shohreh Kasaei. "Human action categorization using discriminative local spatio-temporal feature weighting." *Intelligent Data Analysis* 16, No. 4, (2012), 537-550, doi: 10.3233/IDA-2012-0538.
  63. An, Feng-Ping. "Human action recognition algorithm based on adaptive initialization of deep learning model parameters and support vector machine." *IEEE Access* 6, (2018), 59405-59421, doi: 10.1109/ACCESS.2018.2874022.
  64. Soomro, Khurram, Amir Roshan Zamir, and Mubarak Shah. "UCF101: A dataset of 101 human actions classes from videos in the wild." arXiv preprint arXiv:1212.0402 (2012).
  65. Karpathy, Andrej, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. "Large-scale video classification with convolutional neural networks." In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, 1725-1732. 2014.
  66. Hou, Rui, Amir Roshan Zamir, Rahul Sukthankar, and Mubarak Shah. "Damn-discriminative and mutually nearest: Exploiting pairwise category proximity for video action recognition." In European Conference on Computer Vision, 721-736. Springer, Cham, 2014.
  67. Boyraz12, Hakan, Syed Zain Masood13, Baoyuan Liu, Marshall Tappen12, and Hassan Foroosh. "Action recognition by weakly-supervised discriminative region localization." (2014).
  68. Kihl, Olivier, David Picard, and Philippe-Henri Gosselin. "A unified framework for local visual descriptors evaluation." *Pattern Recognition* 48, No. 4, (2015), 1174-1184, doi: 10.1016/j.patcog.2014.11.013.
  69. Peng, Xiaojiang, and Cordelia Schmid. "Multi-region two-stream R-CNN for action detection." In European conference on computer vision, pp. 744-759. Springer, Cham, 2016.
  70. Chang, Xiaojun, Yao-Liang Yu, and Yi Yang. "Robust top-k multiclass SVM for visual category recognition." In Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 75-83. 2017.
  71. Hua, Michelle, Mingqi Gao, and Zichun Zhong. "SCN: Dilated silhouette convolutional network for video action recognition." *Computer Aided Geometric Design* 85, (2021), 101965, doi: 10.1016/j.cagd.2021.101965.
  72. dos S Silva, Francisco H., Gabriel M. Bezerra, Gabriel B. Holanda, J. Wellington M. de Souza, Paulo AL Rego, Aloísio V. Lira Neto, Victor Hugo C. de Albuquerque, and Pedro P. Rebouças Filho. "A novel feature extractor for human action recognition in visual question answering." *Pattern Recognition Letters* 147, (2021), 41-47, doi: 10.1016/j.patrec.2021.04.002.
  73. Leyva, Roberto, Victor Sanchez, and Chang-Tsun Li. "Compact and low-complexity binary feature descriptor and Fisher vectors for video analytics." *IEEE Transactions on Image Processing* 28, No. 12 (2019): 6169-6184, doi: 10.1109/TIP.2019.2922826.
  74. Sahoo, Suraj Prakash, and Samit Ari. "On an algorithm for human action recognition." *Expert Systems with Applications* 115, (2019), 524-534, doi: 10.1016/j.eswa.2018.08.014

---

**Persian Abstract**

---

**چکیده**

در این مقاله، یک دسته بند جدید مبتنی بر ماشین بردار پشتیبان دوقلو خطی برای مواجهه با مشکل نواحی غیرقابل دسته بندی در مسائل دسته بندی چندکلاسه ارائه شده است. الگوریتم پیشنهادی با عنوان ماشین بردار پشتیبان دوقلو احتمالاتی روی مدل حاصل از ماشین بردار پشتیبان دوقلو خطی یک خروجی پیوسته و احتمالاتی تولید می کند. این الگوریتم میتواند مشکل نواحی غیرقابل دسته بندی را با بکارگیری یک تابع عضویت برطرف کرده، اثرات نامطلوب داده های نویزی را کاهش دهد. کارایی الگوریتم پیشنهادی به کمک چندین مجموعه داده شامل دادگان تشخیص رفتار انسان ارزیابی شده است. نتایج بیانگر کارایی بهتر الگوریتم پیشنهادی نسبت به روش های مشابه است.

---